AIAA Guidance, Navigation, and Control Conference and Exhibit
15 - 18 August 2005, San Francisco, California

AIAA 2005-6191

Path Planning by Unmanned Air Vehicles for Engaging an Integrated Radar Network

Reid A. Larson, 1st Lt, USAF*
Air Force Research Laboratory, Wright-Patterson AFB, OH 45433

Meir Pachter†
Air Force Institute of Technology, Wright-Patterson AFB, OH 45433

Mark J. Mears‡
Air Force Research Laboratory, Wright-Patterson AFB, OH 45433

A growing concept in the field of unmanned air vehicles (UAVs) is the idea of using a team of cooperating vehicles to participate in electronic countermeasures, defined here as jamming or deception techniques. A UAV may be tasked to engage a radar using noise jamming to mask its radar return or that of another vehicle. Similarly, a UAV may be assigned to deceive a radar by directing a delayed signal toward the victim radar, which has the effect of producing a radar phantom perceived by the radar as an object at a false range and/or bearing. Previous work focused on generating a set of waypoints for the UAV to follow in order for the countermeasures to be successful. This paper addresses the path planning required to meet the temporal, spatial, and UAV flight dynamics constraints associated with employing these electronic countermeasures, especially between jamming and deception activities. The UAVs are assigned simplified flight dynamics and performance constraints in two-dimensions, assuming constant altitude flight over a flat-surfaced earth. All tracking radars are given simplified detection properties. A single UAV is provided a pre-determined series of "goal positions." The goal positions may lie along a countermeasure's pre-planned course or they may be established such that the UAV moves from the final waypoint of one countermeasure to the starting point of the next countermeasure. Therefore the UAV must autonomously navigate to a given goal position, subsequently perform a simple, associated task (countermeasure, if required), then navigate to the next goal position in the series. The UAVs will be required to arrive at these waypoints with a specific state, depending on the task at hand. Algorithms for optimal autonomous navigation of this nature were formulated to effectively guide the UAVs to their goal positions to meet the necessary temporal and spatial requirements. Simulations were generated to test the path-planning and control strategies given UAV/radar network scenarios, and overall UAV navigational performance in each simulation was analyzed.

Nomenclature

$\overline{\mathbf{S}}$	State vector
$\overline{\mathbf{u}}$	Control vector
x	x-position
y	y-position
V	Velocity
ψ	Heading
t	Time

*Control Systems Engineer, Air Vehicles Directorate, AFRL/VACA
†Professor, Dept. of Electrical Engineering, AFIT/ENG
‡Ph.D., Control Systems Engineer, Air Vehicles Directorate, AFRL/VACA

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
AUG 2005		**00-00-2005 to 00-00-2005**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Path Planning by Unmanned Air Vehicles for Engaging an Integrated Radar Network	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Air Force Research Laboratory,Air Vehicles Directorate,Wright Patterson AFB,OH,45433**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**16**	

ΔT Time-step (also Δt)
i Time index
r Radial distance
θ Angular position (polar coordinates)

Subscripts
u Pertaining to a UAV
r Pertaining to a radar
p Pertaining to a radar phantom
g Pertaining to a goal position

I. Introduction

A growing concept in the field of unmanned air vehicles (UAVs) is the idea of using a team of cooperating vehicles to participate in electronic countermeasures against a radar network. Here, electronic countermeasures are defined as using either jamming techniques, deception techniques, or a combination of the two to assist in meeting mission objectives. UAVs may employ noise jamming by directing energy toward a radar to mask its return or that of a fellow team member. Radar deception involves detecting a radar signal and subsequently producing and directing a delayed signal with the same characteristics back toward a victim radar to create a false target (or phantom) with false range and/or bearing information. Depending on the number of UAVs in the team and the number of radars in the network, the UAVs may be able to employ different countermeasures simultaneously. A characteristic of the electronic countermeasure activities is that they require the UAVs to follow time-critical, directionally-dependent trajectories with tight constraints in order to be successful, from the start of the activity to the very end. It is absolutely necessary that UAVs are able to control their movements during countermeasure tasks as well as navigate in a coordinated fashion en route to the subsequent tasking. UAV path-planning for radar countermeasures themselves has been developed through previous work.[1] In this paper, we will look at two UAV navigation approaches to maneuver a UAV from one tasking to the next.

This paper begins with a discussion of the UAV and radar network engagement (Section 2). The UAVs employ electronic countermeasures in the form of radar deception by generating phantom targets and radar jamming by shrinking radar detection circles; both which require the UAVs to follow certain trajectories for success. The temporal and spatial constraints required for the countermeasure tasks require the UAVs to rendezvous prescribed states between the ending of one activity and the beginning of the subsequent activity. The focus of this research becomes maneuvering to these so-called goal-states. The first path-planning technique takes advantage of the geometry between a UAV's initial position and its goal position and calculates a trajectory within tight flight dynamics (Section 3). An algorithm is presented for employing the geometry-based path-planning in practice and an example of the algorithm is simulated for analysis. A second path-planning strategy uses a dynamic optimization approach to reach goal states and allows the UAV flexibility in varying its attitude to reach a goal given tight temporal constraints while simultaneously avoiding radar sites as required (Section 4). The optimization strategy is tested in computer simulation with two scenarios and the results are presented. The paper concludes with insights into the path-planning strategies presented and discusses future thrusts in this research area.

II. Radar Network Engagement

In this section, we briefly discuss two countermeasures that UAVs use against a radar network; deception and jamming. The network has j radars, and the j^{th} radar is positioned in two-dimensional space at $[x_{rj}, y_{rj}]$. Each radar has a detection circle of radius Rad_{rj} centered about the radar site. An object (UAV) within a circle is considered "detected" by that radar. Entering a region of overlapping detection circles will cause more than one radar to detect the object. Objects outside a radar's detection circle are not detected by that radar. Given these simple radar attributes, we are now prepared to discuss the deception and jamming scenarios.

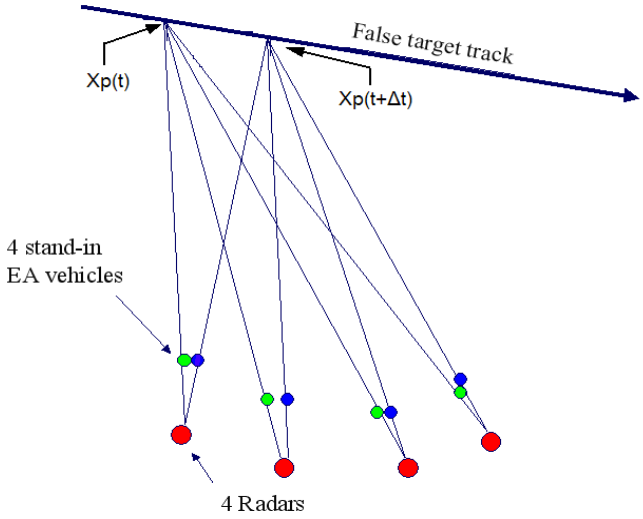

Figure 1. Four UAVs generating a single, coherent phantom track using range delay deception.

A. Deception Scenario

One countermeasure technique that a UAV (or UAV team) can employ against a radar network is using range delay deception to generate coherent radar phantom tracks. The discussion contained in this sub-section is briefly summarized from Pachter et.al.[1] In range delay deception, a pulse from a radar is detected by a *stealthy* UAV and retransmitted back to the targeted radar. This causes the victim radar to calculate an erroneous target range. Multiple tracking radars can cooperate in an integrated defense network by correlating target tracks to perceive this deception. In view of this, multiple UAVs can further cooperate in deceiving the networked radars by generating a single coherent phantom track. Figure 1 illustrates an ideal phantom track scenario. In this example, there are four radars that share track files around the network. There are four UAVs, one assigned to each radar. At time t, the UAVs are in the radars' line of sight to the phantom target's position $\overline{\mathbf{x}}_p(t)$. The radar pulses are individually delayed by the UAVs so that the perceived range vectors all intersect at $\overline{\mathbf{x}}_p(t)$. The UAVs are then repositioned to continuously remain in the radars' line of sight at $t+\Delta t$, in which case each radar again registers a perceived return at $\overline{\mathbf{x}}_p(t+\Delta t)$. The sequence of registered false target returns at a perceived speed(s) and heading(s) constitutes the false target track, or phantom track. Since each of the radars confirms the other's target track, the track is considered valid by the whole network. The phantom target is invariably placed in the detection circle of the victim radars.

UAV trajectories for the phantom track deception problem can be determined by either (1) solving the "inverse problem," i.e., calculate a UAV trajectory based on a pre-determined phantom target trajectory; or (2) solving the "forward problem," i.e., solution by incrementally stepping forward in time given phantom trajectory data. The kinematic diagram of the engagement from which the solutions of the inverse and forward problems are determined is shown in Figure 2.

1. The Inverse Problem

It is often required to synthesize a UAV trajectory on the time interval $t_0 \leq t \leq t_f$, for a specified phantom trajectory; hence we solve the "inverse problem." Suppose that the phantom target's required path is given by $[x_p(t), y_p(t)]$. The track is converted to polar coordinates (for mathematical convenience) and is specified by $[R_p(t), \theta(t)]$, $t_0 \leq t \leq t_f$. It is required to synthesize the UAV's trajectory $R_u(t)$ (and/or $R_u(\theta)$). Since the UAV is required to remain in the radar's line of sight, it is easy to see that $\theta(t)$ must be equivalent for

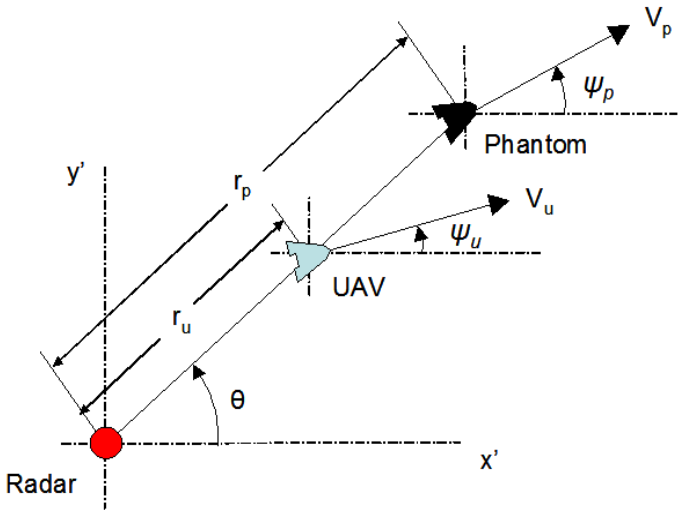

Figure 2. Kinematics of the UAV phantom target deception. A local set of reference axes is placed with the origin located at a given radar's position in the global coordinate system. The UAV's kinematic representation (position, velocity, and heading) will be used throughout this paper.

both the phantom and the UAV. By introducing the following dimensionless parameters:

$$r_p(t) = \frac{R_p(t)}{R_p(t_0)}; \;\; r_u(t) = \frac{R_u(t)}{R_p(t_0)},$$

and relating the system kinematics (Figure 2), we can readily obtain the following differential equations:

$$\dot{r}_u(t) = \sqrt{1 - (\frac{d\theta}{dt}(t))^2 r_u^2(t)} \;, \quad r_u(t_0) = r_{u0} \;, \quad t_0 \leq t \leq t_f \tag{1}$$

and

$$\dot{r}_u(t) = -\sqrt{1 - (\frac{d\theta}{dt}(t))^2 r_u^2(t)} \;, \quad r_u(t_0) = r_{u0} \;, \quad t_0 \leq t \leq t_f \tag{2}$$

Equations (1) and (2) are solved separately for the cases in which the UAV's range from the victim radar monotonically increases or decreases, respectively. $d\theta/dt$ is known based on the phantom target's pre-determined trajectory. The radicand quantity in each equation above is essentially a switching function, i.e. $s(t) \equiv 1 - (\frac{d\theta}{dt}(t))^2 r_u(t)^2$. Solutions must be obtained from equations (1) and (2) separately as required in order to keep the quantity $\sqrt{s(t)}$ from becoming imaginary. The polar UAV trajectory, $r_u(t)$, $\theta(t)$, is converted back to a standard Cartesian coordinate system. The conversion then produces the UAV state $\overline{\mathbf{S}}_u = [x_u(t), y_u(t), V_u(t), \psi_u(t)]^T$ (see Figure 2) from which a flight plan for the UAV is then generated for the specific tasking at hand.

2. The Forward Problem

The solution of the inverse problem may not be desirable in certain instances. For example, the UAV course may require maneuvering in an area considered "off-limits" or the solution may violate its flight constraints. Solution of the "forward problem" over specific time intervals may be required in these cases. Here, an arbitrary trajectory and velocity profile may be pre-determined for either the phantom track or the UAV. Depending on which is provided, the other object's flight plan will be synthesized with respect to time. This case is solved by numerically stepping forward in time and is well-suited to transition to the case of multiple UAVs cooperatively generating a single, coherent phantom track in a radar network. Using the kinematic

geometry of Figure 2, a numerical formula (via the rectangle rule) can be easily determined relating the UAV trajectory to the phantom trajectory:

$$V_u(i) = \frac{x_p(i+1)y_u(i) - y_p(i+1)x_u(i)}{(y_p(i+1)\cos\psi_u(i) - x_p(i+1)\sin\psi_u(i))\Delta t} \tag{3}$$

The time interval $t_0 \leq t \leq t_f$ here is divided into N time steps of length Δt and i represents the i^{th} time step in the interval ($i = 0, 1, ..., N$). A very similar formula can be derived to determine the incremental phantom target velocity $V_p(i)$. In order for equation (3) to be of use, we require knowledge of the position of the phantom target at discrete time increments. We can therefore extract the UAV trajectory by specifying the phantom trajectory over time and assigning flight dynamics to the UAV. The UAV waypoints are specified by $x_u(i)$ and $y_u(i)$, where x and y are the coordinates of an object in the local reference frame x'/y' as shown in Figure 2.

B. Jamming Scenario

A second countermeasure that a UAV may be assigned to is a radar jamming mission. In the context of this paper, a UAV performs a jamming mission to effectively shrink a radar's detection circle. The UAV emits noise energy in the direction of a radar to raise the level of perceived background noise. The radar receives power from a jamming signal which is proportional to $1/(r_u^2)$, where r_u is the radial distance of a UAV jammer from a given radar. Similarly, the received return signal power from an object in a detection circle is proportional to $1/(r^4)$. It is easy to see that there exists a point (i.e., *range*) where the power of an object's radar return will equal the received power from the noise energy generated by the UAV jammer. This point is known as the "burn-through" range and thus provides a lower-bound for which the UAV jamming will be effective.[2,3]

The objective of the UAV jamming mission is to shrink the detection circle of one or more radars to a level which would mask another object's return, thus allowing the object to pass by the radar(s) undetected. It is conceivable that a UAV would be assigned to jam one or more radars over a certain time interval, in which case a flight path is specified over the interval that allows the UAV to meet this objective. The jamming flight paths are not necessarily as specialized as those for deception missions, primarily because the UAV jammer is not tied to another object's kinematics as is the case in the deception scenario with the phantom target.

C. En Route Activities

The last piece of the puzzle in the radar engagement scenarios is the ability to navigate the UAV between countermeasure activities, which is also the main focus of this paper. The success of the countermeasure taskings are dependent on the ability to follow highly directional paths. There are many reasons for this, including deceptor/jammer antenna directionality and UAV flight performance limitations. At the end of a countermeasure tasking, the UAV will be at a given state (position, velocity, and heading). The UAV will need to autonomously navigate to the the prescribed initial state of the next countermeasure activity in light of temporal, spatial, and UAV flight constraints.

There are a few important reasons for expanding the UAV autonomous navigation algorithms used in the described electronic countermeasures. First, electronic countermeasure missions such as this can be very time-critical; that is, there are often tight temporal constraints not only from one countermeasure task to the next, but also within the countermeasure taskings themselves. Further, we would like the UAVs to not only reach positions from which they can launch the countermeasure on time, but also have them positioned so the vehicle is aimed properly along the prescribed course so the tasking can begin immediately. For example, it does little good to have a UAV arrive to a point in space only to have to turn completely around in order to follow the course as was intended. This wastes valuable time that the UAVs may need to meet the mission requirements. For this reason, it is vital that the UAVs can plan their trajectories autonomously between countermeasure taskings. The next two sections address the problem of generating these trajectories. The first section outlines an algorithm which will plan a path for the UAVs in minimum time given specific vehicle constraints (i.e. heading and turn-rate requirements). The second section discusses an optimization approach for the path planning that gives the UAV considerable flexibility in adjusting its flight in order to arrive on-time and on course.

III. Geometry-Based Path Planning Strategy

This section takes an initial geometric-based look at navigational requirements given only spatial constraints; that is, we assume that there is no limitation to the time required to complete a tasking but their is a requirement on the final position and heading of the vehicle.

A. UAV Flight Dynamics and Goal Positions

The UAV is assigned simple two-dimensional dynamics. In discrete form, a single vehicle has the state $\overline{\mathbf{S}}_u$ and follows the equations of motion as shown:

$$\overline{\mathbf{S}}_u(i) = \begin{bmatrix} x_u(i) \\ y_u(i) \\ V_u(i) \\ \psi_u(i) \end{bmatrix} \; ; \; \overline{\mathbf{S}}_u(i+1) = \begin{bmatrix} x_u(i) + V_u(i)\Delta T cos\psi_u(i) \\ y_u(i) + V_u(i)\Delta T sin\psi_u(i) \\ V_u + \Delta V_u(i) \\ \psi_u(i) + \Delta\psi_u(i) \end{bmatrix}; \quad i = 0, 1, ..., N \tag{4}$$

Note the equations of motion for the geometry-based strategy are adjusted to suit a constant velocity vehicle; thus $\Delta V_u = 0$. Further, the change in heading between time steps, $\Delta\psi_u(i)$ is either zero or plus/minus a constant value. Thus, the UAV is a constant velocity, constant turn-rate vehicle.

The UAV will have a set of pre-planned countermeasure tasks that it must perform during the course of its mission. Each countermeasure tasking will require the vehicle to reach a specific waypoint with an accompanied state, i.e., a "goal position." We denote the k^{th} goal position with the subscript gk and assign the goal state as $\overline{\mathbf{S}}_{gk} = [x_{gk}, y_{gk}, V_{gk}, \psi_{gk}]^T$. The goal positions generally correspond to the initial state for each countermeasure trajectory, although they could correspond to intermediate states as well. The goal positions may be stationary or dynamic, following a moving track (that is, the goal may be a function of time in two-dimensional space). We only consider stationary goals in this paper.

B. Path Planning; Minimum Time to Goal

Consider the maneuver required to move a UAV between two consecutive waypoints (initial point to the goal position), each with set states in two-dimensional space. Refer to Figure 3. The UAV follows a constant velocity, constant turn-rate, two-dimensional Dubin's Car model[4] in which the vehicle has an initial (or "original") position given by $p_o = [x_{uo}, y_{uo}]$ and initial heading ψ_{uo}. A new set of reference axes is generated by aligning the y'-axis with the velocity vector at the initial point and setting the origin at p_o. From this information, the UAV computes two possible turn circles for the initial position; one in the clockwise direction and the other in the counter-clockwise direction. The final (goal) position and heading, $p_f = [x_{uf}, y_{uf}]$ and φ_f, are placed in the reference axes as shown. The UAV computes two additional turn circles precisely as with the initial position and heading. Thus, there are four turn circles, and each circle is referenced by its center point and polarity: O_{cco} (original position counterclockwise), O_{co} (original position clockwise), O_{ccf} (final position counterclockwise), and O_{cf} (final position clockwise). The center points are assumed to be translated into the new reference coordinate system. The UAV, however, must process position and heading information in both the local reference coordinate system and the global coordinate system (i.e., the surface map).

The UAV must move from one turn circle to the next without violating its turning constraints in order to reach the final goal position and heading. There are four possible maneuvers with respect to the turn circles to begin and end on (initial circle polarity listed first, final circle polarity second): (a) co/cf, (b) co/ccf, (c) cco/cf, and (d) cco/ccf. Each of the four combinations are feasible for the UAV to choose and utilize, however each combination is evaluated and ultimately the maneuver that will bring the UAV to the goal in the minimum time is chosen. For a constant velocity, constant turn-rate UAV as here, this corresponds to choosing the maneuver with the minimum path length. Further, from these combinations there are three distinct sub-cases that must be addressed to determine the manner the UAV moves from one turn-circle to the next. Each sub-case is dependent on the distance between centers of the initial and final turn circles and not all three are guaranteed to be attainable. The variable d_x represents this distance between the centers of the initial and final turn circles, where the subscript x is replaced by a, b, c, or d for each of the four combinations listed above (the UAV must calculate d for each possible case).

Consider the first sub-case in which the turning circles about p_o and p_f are completely disjoint, that is, the distance between centers of the clockwise and counterclockwise circles are $d_x > 2$. Now examine the

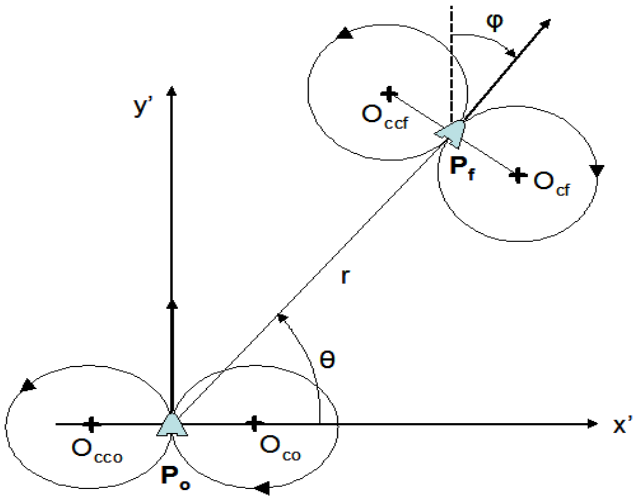

Figure 3. Diagram showing UAV turn circles associated with initial and final positions given associated states. This diagram is shown in the local transformed reference axes. r and θ are used only to calculate distances between turn circles.

geometry of Figure 4. The geometry shows two turn circles and four possible paths, each stemming from the polarity of the initial position turn circle and the final position turn circle. The circle on the left is one of the turn circles tangent to the initial position; the circle on the right corresponds to one of the two turn circles tangent to the final position. There are four tangent paths, represented by dotted lines, that show potential UAV paths to move from one turn circle to the next. Depending on the polarity of each turn circle, two of the four paths are feasible and two of the four paths will be infeasible. For example, if the initial turn circle has clockwise polarity (co) and the final turn circle has counter-clockwise polarity (ccf), the the two feasible paths between circles are $\overline{BF}$ and $\overline{DC}$. Similarly, if the initial turn circle has counterclockwise polarity (cco) and the final turn circle has counterclockwise polarity (ccf), then the two feasible paths between circles are $\overline{EA}$ and $\overline{A'E'}$. The feasible/infeasible paths can be determined for the other two polarity combinations in the same manner.

The second sub-case to consider is that in which two turn circles are conjoined, i.e., $d_x < 2$. Here, the intersecting tangent paths $\overline{BF}$ and $\overline{CD}$ are always infeasible, and the parallel tangent paths $\overline{AE}$ and $\overline{A'E'}$ are feasible only if the turn circles have the same polarity.

In the third sub-case, the initial turn circle and the final turn circles have the same polarity and $d_x \leq 4$. If the initial position and the final position are in close proximity, it may be advantageous to include a third tangent turning circle in place of a straight path outlined in the first and second sub-cases. This sub-case is visually depicted in Figure 5, in which tangent circles are drawn to both the initial and final clockwise turn circles. The opposite scenario (counterclockwise initial and final turn circles) is analagous.

Algorithm

The algorithm to take a constant-velocity, constant turn-rate UAV from its initial position and heading to its goal position and heading is now presented. u^* and ℓ^* represent the minimum-time optimal flight control sequence and optimal associated path length, respectively. To determine the value of ℓ^*, one must use the control sequence u^* and the UAV equations of motion. u^*, depending on the scenario, is gathered from the geometry in Figures 4 and 5. The UAV commands heading changes between initial and final positions as follows: "+1" denotes a clockwise turn; "-1" denotes a counterclockwise turn; "0" denotes no turn (straight path). For example, a typical control sequence using the case in Figure 4 may be represented by $u^* = [+1_A, 0_E, +1_{pf}]$. This is interpreted as piloting the UAV in a clockwise turn from the initial position p_o to point A, followed by a straight path from point A to point E, followed by a clockwise turn from point E to point p_f. The value of ℓ^* is then calculated based on flight dynamics and the distance between turn circle

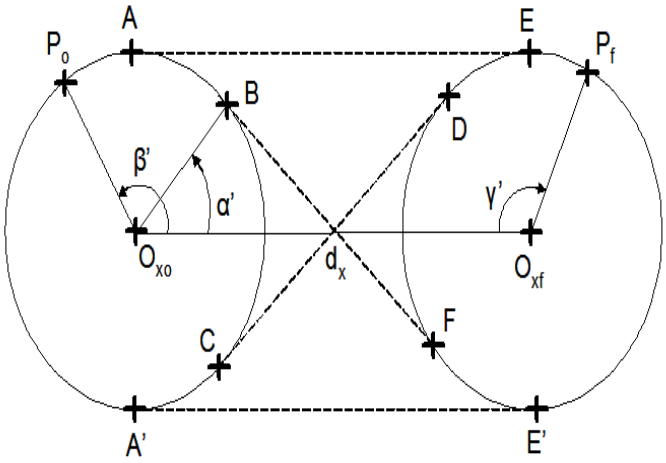

Figure 4. Tangent flight paths between turn circles; the polarity of each turn circle determines which two of the four paths are feasible. The polarity subscripts c (clockwise) or cc (counter-clockwise) are substituted for the subscript x in the turn-circle centers as is appropriate.

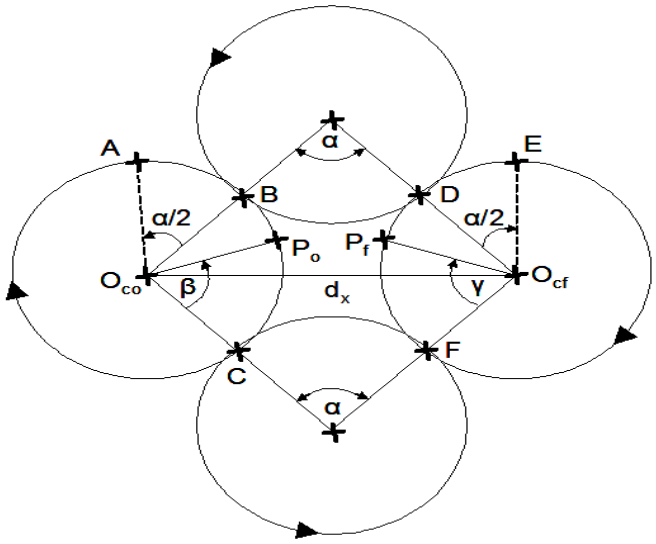

Figure 5. Tangent flight paths between turn circles for a clockwise/clockwise turn circle pair (diagram exactly reversed for a counter-clockwise pair). In this sub-case, a UAV traverses three turn circles to reach its final position and heading.

centers. In this manner an optimal, minimum-time path can be determined using the following algorithm.

1. Determine initial position and heading of the UAV and the final goal position and heading of the UAV in the global reference frame.

2. Establish reference axes x' and y' at UAV's current position, with the y' axis aligned with the current vehicle heading (see Figure 3). Compute the location of turn circle centers O_{co}, O_{cco}, O_{cf}, and O_{ccf} in the new local reference axes. Calculate the values r and θ, according to Figure 3.

3. Calculate d_a, d_b, d_c, and d_d, the distances between centers of the turn circles corresponding to the four turn sequence combinations.

4. Evaluate the time to reach the goal position with heading for case (a) co/cf.

   a. If $d_a > 4$: Compute β' and γ', according to the geometry in Figure 4. Use control $u_a^* = [+1_A, +0_E, +1_{pf}]$ and calculate path length ℓ_a^* using this control sequence.

   b. If $d_a \leq 4$: Compute β, γ, and α according to the geometry in Figure 5. Choose control u_a^* based on the table below, and calculate associated path length ℓ_a^*:

   For $0 < \beta \leq \pi - \alpha$

$$u_a^* = \begin{cases} [+1_C, -1_F, +1_{pf}] & ; & 0 < \gamma \leq \pi - \alpha & ; & 0 < \alpha \leq \pi \\ [+1_C, -1_F, +1_{pf}] & ; & \pi - \alpha < \gamma \leq \pi - \frac{\alpha}{2} & ; & \pi/2 < \alpha \leq \pi \\ [+1_B, -1_D, +1_{pf}] & ; & \pi - \alpha < \gamma \leq \pi - \frac{\alpha}{2} & ; & 0 < \alpha \leq \pi/2 \\ [+1_C, -1_F, +1_{pf}] & ; & \pi - \frac{\alpha}{2} < \gamma \leq 2\pi & ; & \overline{\alpha} < \alpha \leq \pi \\ [+1_A, +0_E, +1_{pf}] & ; & \pi - \frac{\alpha}{2} < \gamma \leq 2\pi & ; & 0 < \alpha \leq \overline{\alpha} \end{cases}$$

   For $\pi - \alpha < \beta \leq \pi - \frac{\alpha}{2}$

$$u_a^* = \begin{cases} [+1_C, -1_F, +1_{pf}] & ; & 0 < \gamma \leq \pi - \alpha & ; & \pi/2 < \alpha \leq \pi \\ [+1_B, -1_D, +1_{pf}] & ; & 0 < \gamma \leq \pi - \alpha & ; & 0 < \alpha \leq \pi/2 \\ [+1_B, -1_D, +1_{pf}] & ; & \pi - \alpha < \gamma \leq \pi - \frac{\alpha}{2} & ; & 0 < \alpha \leq \pi \\ [+1_B, -1_D, +1_{pf}] & ; & \pi - \frac{\alpha}{2} < \gamma \leq 2\pi & ; & 0 < \alpha \leq \pi \end{cases}$$

   For $\pi - \frac{\alpha}{2} < \beta \leq 2\pi$

$$u_a^* = \begin{cases} [+1_C, -1_F, +1_{pf}] & ; & 0 < \gamma \leq \pi - \alpha & ; & \overline{\alpha} < \alpha \leq \pi \\ [+1_A, +0_E, +1_{pf}] & ; & 0 < \gamma \leq \pi - \alpha & ; & 0 < \alpha \leq \overline{\alpha} \\ [+1_B, -1_D, +1_{pf}] & ; & \pi - \alpha < \gamma \leq \pi - \frac{\alpha}{2} & ; & 0 < \alpha \leq \pi \\ [+1_A, +0_E, +1_{pf}] & ; & \pi - \frac{\alpha}{2} < \gamma \leq 2\pi & ; & 0 < \alpha \leq \pi \end{cases}$$

   (Note that $\overline{\alpha}$ is the solution to the transcendental equation $\overline{\alpha} + 2 * sin(\overline{\alpha}/2) = \pi$, a special case that arises from the geometry. This relation will be used later in item (7).)

5. Evaluate the time to reach the goal position with heading for case (b) co/ccf.

   a. If $d_b \leq 2$: Set $\ell_b^* = \infty$.

   b. If $d_b > 2$: Compute β', γ', and α', according to Figure 4. Use control $u_b^* = [+1_B, +0_F, -1_{pf}]$ and calculate path length ℓ_b^* using this control sequence.

6. Evaluate the time to reach the goal position with heading for case (c) cco/cf.

   a. If $d_c \leq 2$: Set $\ell_c^* = \infty$.

   b. If $d_c > 2$: Compute β', γ', and α', according to Figure 4. Use control $u_c^* = [-1_C, +0_D, +1_{pf}]$ and calculate path length ℓ_c^* using this control sequence.

7. Evaluate the time to reach the goal position with heading for case (d)–cco/ccf.

a. If $d_d > 4$: Compute β' and γ', according to Figure 4. Use control $u_d^* = [-1_{A'}, +0_{E'}, -1_{pf}]$ and calculate path length ℓ_d^* using this control sequence.

b. If $d_d \leq 4$: Compute β, γ, and α according to the geometry in Figure 5. Choose control u_d^* based on the table below, and calculate associated path length ℓ_d^*:

For $0 < \beta \leq \pi - \alpha$

$$u_d^* = \begin{cases} [-1_C, +1_F, -1_{pf}] & ; & 0 < \gamma \leq \pi - \alpha & ; & 0 < \alpha \leq \pi \\ [-1_C, +1_F, -1_{pf}] & ; & \pi - \alpha < \gamma \leq \pi - \frac{\alpha}{2} & ; & \pi/2 < \alpha \leq \pi \\ [-1_B, +1_D, -1_{pf}] & ; & \pi - \alpha < \gamma \leq \pi - \frac{\alpha}{2} & ; & 0 < \alpha \leq \pi/2 \\ [-1_C, +1_F, -1_{pf}] & ; & \pi - \frac{\alpha}{2} < \gamma \leq 2\pi & ; & \overline{\alpha} < \alpha \leq \pi \\ [-1_A, +0_E, -1_{pf}] & ; & \pi - \frac{\alpha}{2} < \gamma \leq 2\pi & ; & 0 < \alpha \leq \overline{\alpha} \end{cases}$$

For $\pi - \alpha < \beta \leq \pi - \frac{\alpha}{2}$

$$u_d^* = \begin{cases} [-1_C, +1_F, -1_{pf}] & ; & 0 < \gamma \leq \pi - \alpha & ; & \pi/2 < \alpha \leq \pi \\ [-1_B, +1_D, -1_{pf}] & ; & 0 < \gamma \leq \pi - \alpha & ; & 0 < \alpha \leq \pi/2 \\ [-1_B, +1_D, -1_{pf}] & ; & \pi - \alpha < \gamma \leq \pi - \frac{\alpha}{2} & ; & 0 < \alpha \leq \pi \\ [-1_B, +1_D, -1_{pf}] & ; & \pi - \frac{\alpha}{2} < \gamma \leq 2\pi & ; & 0 < \alpha \leq \pi \end{cases}$$

For $\pi - \frac{\alpha}{2} < \beta \leq 2\pi$

$$u_d^* = \begin{cases} [-1_C, +1_F, -1_{pf}] & ; & 0 < \gamma \leq \pi - \alpha & ; & \overline{\alpha} < \alpha \leq \pi \\ [-1_A, +0_E, -1_{pf}] & ; & 0 < \gamma \leq \pi - \alpha & ; & 0 < \alpha \leq \overline{\alpha} \\ [-1_B, +1_D, -1_{pf}] & ; & \pi - \alpha < \gamma \leq \pi - \frac{\alpha}{2} & ; & 0 < \alpha \leq \pi \\ [-1_A, +0_E, -1_{pf}] & ; & \pi - \frac{\alpha}{2} < \gamma \leq 2\pi & ; & 0 < \alpha \leq \pi \end{cases}$$

8. Determine minimum length path to reach final position and final heading, that is, choose $min(\ell_a^*, \ell_b^*, \ell_c^*, \ell_d^*)$ and the its associated control scheme u^*.

C. Simulation Example

Here, we present an example using the path-planning algorithm described. The UAV flies over a flat, two-dimensional map ranging from ± 10 units in both the east/west directions (x-axis) and the north/south directions (y-axis). Three radars are located in the map-space. The first radar is located at $[x_{r1}, y_{r1}] = [0, 0]$ with detection circle radius of $Rad_{r1} = 3.0$; the second radar is located at $[x_{r2}, y_{r2}] = [-7, -7]$ with detection circle radius of $Rad_{r2} = 6.0$; the third radar is located at $[x_{r3}, y_{r3}] = [+7, +7]$ with detection circle radius of $Rad_{r3} = 6.0$. The UAV is assigned a constant velocity of 1.0 (unit distance/unit time) and a turn rate of 1.0 (rad/unit time). The initial state of the UAV is $[x_{uo}, y_{uo}, V_{uo}, \psi_{uo}]^T = [-5, +5, 1.0, 0°(\pi/180°)]^T$ and the UAV must reach the final goal position $[x_{uf}, y_{uf}, V_{uf}, \psi_{uf}]^T = [+5, -5, 1.0, 180°(\pi/180°)]^T$ to prepare for a subsequent tasking.

The algorithm in the preceding subsection was programmed into a MATLAB simulation. In Figure 6, we see a plot of the map-space, the three radar sites, and the UAV's initial position and goal positions with accompanying turn circles. The algorithm calculated the minimum path length $\ell_a^* = [0.6747, 12.8062, 2.4669]$ (total distance traveled 15.9478 units) and the associated control sequence $u_a^* = [+1, 0, +1]$ as the desired flight path. The UAV is represented by a small green triangle. The UAV is plotted at each time step to easily view the total path traversed in the map-space (note that each time unit is subdivided into five sub-units (time steps) for simulation use).

The discussion of this section and the example shown in Figure 6 allow us to draw a few conclusions about this path-planning strategy. First, the strategy is very simple to implement and could accommodate variable speed and turn-rates between turn circles with modifications to the formulation and algorithm. Further, we are guaranteed that a trajectory and command sequence will be generated (quickly) regardless of initial and final goal positions. There are two main disadvantages to using this type of algorithm. The most important disadvantage to this application is that the algorithm does not allow for detection circle avoidance when desired. In Figure 6, we see the UAV traverses through a detection circle for a considerable

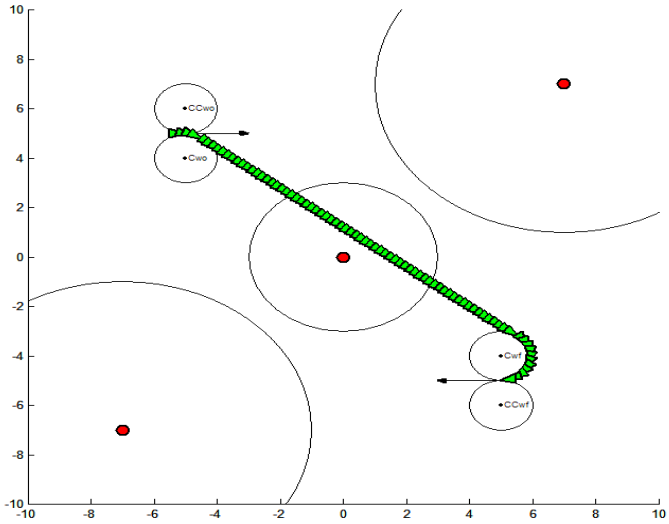

Figure 6. UAV trajectory from initial position to a single goal position using the geometry-based algorithm presented.

period of time in order to reach its goal. This would most likely not be acceptable, especially given that this path planning is performed between countermeasure activities. Ideally, we would like the UAV to plot its own path around the radar site and still reach the goal position as required. The second disadvantage is that while the algorithm is designed as a minimum-time approach, there is almost no control on the arrival time. In certain instances, the UAV may need to control its flight path to arrive at a goal state at a precise time to coordinate with other vehicles in a given mission. Flying directly to a goal just to reach it in minimum time may be detrimental to the subsequent tasking.

IV. Optimization-Based Path Planning Approach

Perhaps a more practical approach than the geometric-based path planning strategy is one using dynamic optimization techniques. The dynamic optimization approach has the distinct advantage that allows the UAV to navigate to goal positions autonomously by varying its velocity and turn-rate throughout the trajectory while simultaneously avoiding other radars and arriving at the goal state within a given time window. This is a large improvement from the geometry-based strategy, and the formulation is described within this section.

A. Discrete Dynamic Optimization

This path planning strategy involves solution of a discrete dynamic optimization problem with temporal and spatial constraints at time t_f (time t_f here is the ideal time that the UAV should arrive at the goal position). The time interval $t_0 \leq t \leq t_f$ is divided into N discrete time steps, and the index i indicates an individual time step such that $i = 0, 1, ..., N$. This formulation is not as elegant as the closed-form, continuous formulation, but it is more suited for computer simulation and is flexible to many different scenarios. The optimization process is performed by minimizing $\overline{J}$, the objective function adjoined with constraints, in Mayer form below[5,:6]

$$\min_{\overline{\mathbf{S}}_u} \overline{J}_u = \phi_u + \nu_u^T \Psi_u + \sum_{i=0}^{N-1} \lambda_u(i+1)\{f[\overline{\mathbf{S}}_u(i), \overline{\mathbf{u}}_u(i), i] - \overline{\mathbf{S}}_u(i+1)\} + \lambda_u^T(0)[\overline{\mathbf{S}}_{u0} - \overline{\mathbf{S}}_u(0)] \tag{5}$$

f represents a vector of functions (i.e. UAV equations of motion and performance cost), λ and ν are vectors of Lagrange multipliers and final state constraint parameters, and $\phi = (N\Delta T - t_f)^2$ where $N\Delta T$ is the solution final time and t_f is the prescribed final time. We define the initial, i^{th} time step, and final augmented UAV

states as:

$$\overline{\mathbf{S}}_u(0) = \begin{bmatrix} x_u(0) \\ y_u(0) \\ V_u(0) \\ \psi_u(0) \\ \cdots\cdots \\ C(0) \end{bmatrix} ; \ \overline{\mathbf{S}}_u(i) = \begin{bmatrix} x_u(i) \\ y_u(i) \\ V_u(i) \\ \psi_u(i) \\ \cdots\cdots \\ C(i) \end{bmatrix} ; \ \overline{\mathbf{S}}_u(N) = \begin{bmatrix} x_u(N) \\ y_u(N) \\ V_u(N) \\ \psi_u(N) \\ \cdots\cdots \\ C(N) \end{bmatrix}$$

The UAV controls are given as

$$\overline{\mathbf{u}}(i) = \begin{bmatrix} \Delta V_u(i) \\ \Delta \psi_u(i) \end{bmatrix} \tag{6}$$

The constraints vector $f_u(i)$ (UAV equations of motion plus cost) is

$$f_u[\overline{\mathbf{S}}_u(i), \overline{\mathbf{u}}_u(i), i] = f_u(i) = \begin{bmatrix} x_u(i) + V_u(i)\Delta T cos\psi_u(i) \\ y_u(i) + V_u(i)\Delta T sin\psi_u(i) \\ V_u(i) + \Delta V_u(i) \\ \psi_u(i) + \Delta \psi_u(i) \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ C_u(i) + L_u(i) \end{bmatrix} \tag{7}$$

$L_u(i)$ is the performance cost at each time step, and $C_u(i)$ is the cumulative performance cost from all time steps up to time step i. This cost is used for radar avoidance so that the UAV generates a solution that avoids certain radar sites; more specifically, the UAV avoids those threat circles where countermeasures will not be employed. Let $r_{uj}(i) = ||[x_{rj}, y_{rj}]^T - [x_u(i), y_u(i)]^T||_2$ be the distance from the UAV to the j^{th} radar at the i^{th} time step. The cost associated with the j^{th} radar is given as:

$$L_{rj}(i) = \begin{cases} \frac{W_{rj}(Rad_{rj})^2}{[r_{uj}(i)]^2} & , \quad Rad_{rj} \geq r_{uj}(i) \\ 0 & , \quad otherwise \end{cases} \tag{8}$$

W_{rj} is an avoidance coefficient; $W_{rj} = 0$ for those radar circles that need not be avoided; $W_{rj} = 1$ for those radars that must be avoided. At each time step, the total cost for radar circle avoidance is $L_u(i) = \sum L_{rj}(i)$ and the cumulative sum of the avoidance cost is $C_u = \sum_{i=0}^{N} L_u(i)$.

The terminal constraints on the final state, Ψ_u, are given by the vector

$$\Psi_u = \begin{bmatrix} x_u(N) - x_{uf} \\ y_u(N) - y_{uf} \\ V_u(N) - V_{uf} \\ \psi_u(N) - \psi_{uf} \\ \cdots\cdots\cdots\cdots \\ C_u(N) - 0 \end{bmatrix} \tag{9}$$

$\overline{\mathbf{S}}_{uf} = [x_{uf}, y_{uf}, V_{uf}, \psi_{uf}]^T = \overline{\mathbf{S}}_{gk} = [x_{gk}, y_{gk}, V_{gk}, \psi_{gk}]^T$ is the required final state. The Hamiltonian, Euler-Lagrange equations, and transversality condition are given as:

$$\begin{aligned} H(i) &= \lambda(i+1) f_u(i); \\ \lambda^T(N) &= \nu^T \Psi_{u\overline{\mathbf{S}}}; \\ \lambda^T(i) &= \lambda^T(i+1) f_{u\overline{\mathbf{S}}}(i); \\ H_{\mathbf{u}}(i) &= \lambda^T(i+1) f_{u\overline{\mathbf{u}}}(i) = 0; \\ \phi_{\Delta T} + \sum_{i=0}^{N-1} H_{\Delta T}(i) &= 0 \end{aligned} \tag{10}$$

The subscripts $\overline{\mathbf{S}}$, $\overline{\mathbf{u}}$, and ΔT on $f(i)$ and the Hamiltonian refer to the partial derivatives of those expressions with respect to those vectors/variables. These equations allow the UAV to optimize the cost function in equation (5) and solve for a course that will meet the state and temporal constraints. The transversality condition is added to solve for the optimal time step ΔT.

Bryson[5] provides a steepest descent numerical algorithm titled Dynamic Optimization for Minimum Time (DOPT) which works very well given a poor initial guess for a solution (linear interpolation between initial and final states is generally a sufficiently good initial guess). DOPT iteratively solves for the optimal control sequence which in turn is used to generate the optimal state trajectory given the UAV equations of motion. A final solution is obtained when the numerical technique converges such that the optimal control sequence changes minimally within a set tolerance between iterations.

1. Flight Limits Override

It is possible that an optimal control sequence may violate known vehicle performance parameters such as maximum or minimum velocities (V_u), accelerations (ΔV_u), and turn rate ($\Delta\psi_u$). If this is indeed the case, eventually the control commands will hit saturation levels on the real-world UAV that may cause its trajectory to deviate drastically such that the UAV does not hit the final goal state. The UAV's navigational algorithm therefore needs to be proactive to ensure that flight constraints are not violated. Assume the following flight limits apply:

$$\begin{aligned} V_{uMin} \leq V_u \leq V_{uMax}; \\ \Delta V_{uMin} \leq \Delta V_u \leq \Delta V_{uMax}; \\ \Delta\psi_{uMin} \leq \Delta\psi_u \leq \Delta\psi_{uMax} \end{aligned} \tag{11}$$

The numerical technique is modified to directly truncate the velocity state, acceleration control, and turn-rate control at the saturation limits between each iteration, if necessary. This has the effect of perturbing the control and state trajectories toward a direction satisfying these constraints. The perturbations will generally still result in a control sequence satisfying the flight constraints, radar avoidance parameters, and the end-state temporal/spatial constraints, although it may be "sub-optimal" in certain respects. For our study, satisfying flight performance parameters and reaching a goal state within these parameters will be considered an optimal solution by the UAV.

B. Simulations

The dynamic optimization technique was simulated in MATLAB in two separate scenarios, labeled (a) and (b). Scenario (a) is precisely the same as that in the geometric-based simulation example but uses the optimization approach for a comparison. Scenario (b) analyzes a new series of goal positions in a radar network and assesses the solution performance. Neither scenario includes attached countermeasure taskings. Combining both the countermeasure tasks and en route activities together in one complete mission is left as future work.

The UAV flight constraints, each on a per unit time-step basis, are given as:

$$\begin{aligned} 0.25 \leq V_u \leq 2.0; \\ -0.05 \leq \Delta V_u \leq 0.5; \\ -15^\circ \leq \Delta\psi_u \leq +15^\circ \end{aligned}$$

Scenario (a) has radars at the same positions with the same attributes as in the geometry-based example in Section 3.C. Scenario (b) has radars at the following positions with the following attributes:

$$\mathbf{Radars}_{(b)} = \begin{bmatrix} \vdots & & & & \\ x_{rj} & \cdots & -6 & 0 & 6 & 0 \\ y_{rj} & \cdots & -5 & -5 & -5 & 4 \\ Rad_{rj} & \cdots & 2 & 2 & 2 & 6 \\ W_{rj} & \cdots & 1 & 1 & 1 & 0 \\ \vdots & & & & & \end{bmatrix}$$

Scenario (a) has the same initial state and final goal state as the geometry-based algorithm example with the

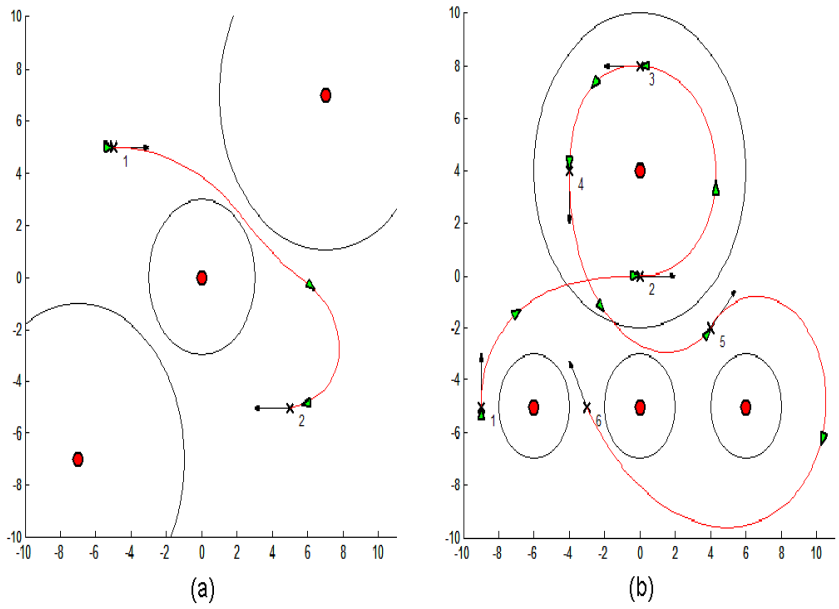

Figure 7. The UAV trajectories for Scenarios (a) and (b) are shown. The UAV moves along the path depicted with the red-line sequentially from one goal to the next. Each goal is shown as located in the map space with an arrow pointing in the prescribed vector's direction.

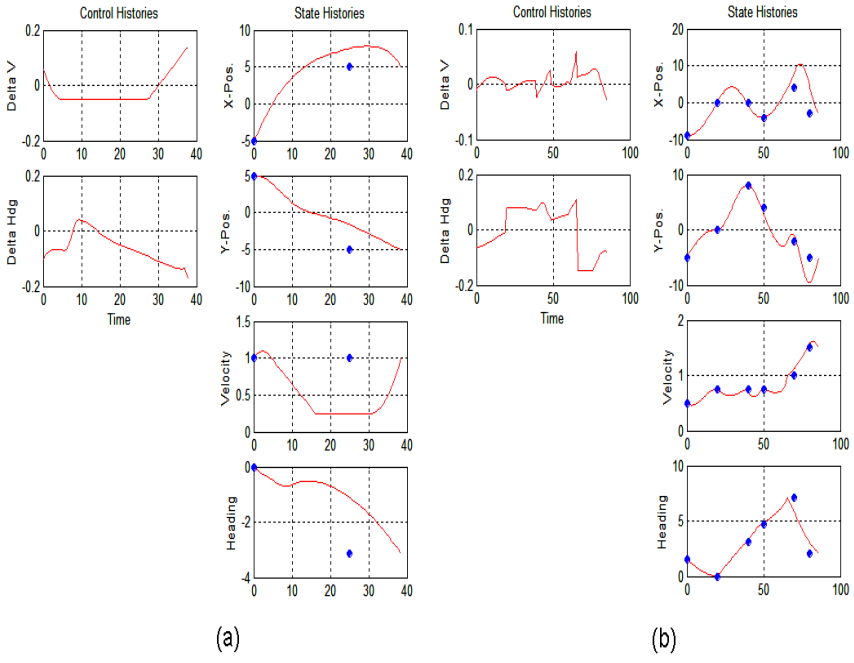

Figure 8. State and control histories for both Scenarios (a) and (b) are shown. The blue dots indicate the desired value for each state variable at the prescribed final time t_f. Note where the UAV reaches saturation limits that must be overcome.

added temporal constraint that the UAV must arrive at the goal at $t_f = 25$. Scenario (b) has the following goal states; the first *column* represents the initial state and the final *row* represents the desired final time to reach the goal state:

$$\mathbf{Goals}_{(b)} = \begin{bmatrix} x_{gk} & \vdots & -9 & 0 & 0 & -4 & 4 & -3 \\ y_{gk} & \vdots & -5 & 0 & 8 & 4 & -2 & -5 \\ V_{gk} & \vdots & 0.50 & 0.75 & 0.75 & 0.75 & 1.00 & 1.5 \\ \theta_{gk} & \vdots & 90^\circ & 0^\circ & 180^\circ & 270^\circ & 45^\circ & 120^\circ \\ \cdots & \vdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ t_f & \vdots & 0 & 20 & 40 & 50 & 70 & 80 \end{bmatrix}$$

The number of time steps N for each leg of the mission was chosen to be two times the total time allowed between legs; that is $N = 2(t_f - t_0)$. Scenarios (a) and (b) were tested using the dynamic optimization strategy; the results are shown in Figures 7 and 8. Figure 7 shows plots of the vehicle trajectories and Figure 8 gives state and control histories for each scenario.

In Figure 7(a), we see the trajectory the UAV takes to reach the goal compared to the trajectory of Figure 6. Notice the UAV here avoids the radar sites and detection circles completely while still arriving at the goal state. The accompanying state and control history in Figure 8(a) shows that the UAV was not able to reach the goal at precisely $t_f = 25$ but rather was delayed about 10 time units. Whether or not the late arrival is problematic is for the mission planner to decide. Also note from Figure 8(a) the points in the control and state histories where saturation limits are hit in the vehicle velocity and acceleration parameters. This demonstrates that the UAV takes into account the saturation limits of its performance parameters when calculating optimal trajectories.

In Figure 7(b), we see the trajectory the UAV takes navigating between six goals. The large radar site at the top is designated as a site that need not be avoided, and thus the UAV navigates freely throughout the detection circle. The three smaller sites at the bottom are designated as sites to avoid, and the UAV makes the effort to avoid them while reaching nearby goals. Figure 8(b) gives the state and control histories for this scenario.

V. Conclusions

This paper covered two path-planning strategies that UAVs could use to assist in radar countermeasure missions, namely those where UAVs use jamming and deception. Given that countermeasure missions require time-critical positioning to be successful, it is necessary for UAVs to autonomously navigate from the final position and state of one countermeasure activity to the initial position and state of the subsequent activity in a coordinated manner. The position and state that the UAV attempts to reach is the UAV's goal state, which may or may not have temporal constraints attached. The first autonomous navigation scheme presented to meet this objective was a geometry-based algorithm that seeks the minimum-time path to reach a goal state given a vehicle with constant velocity and constant turn-rate. The algorithm is useful under specific circumstances, however it is not recommended where tight temporal constraints and radar avoidance are key to mission tasking success. A more practical approach, one using dynamic optimization techniques, yields better results in that it is able to generate feasible solutions where temporal constraints, UAV flight constraints, and radar avoidance are critical factors. The two path-planning strategies could possibly be used in conjunction, especially when the dynamic optimization solution hits saturation limits and the UAV may want a more readily-obtained solution.

Future work will concentrate on combining the three path-planning strategies together (jamming trajectories, deception trajectories, maneuvers between countermeasures) such that the UAV autonomously generates the entire flight plan from start to finish, given a sequence of countermeasure activities and the accompanying temporal constraints. The work will progress toward UAV team interactions, such as path-planning to account for formation flight and contingencies that can occur where multiple UAVs are involved. Finally, further research will incorporate three-dimensional effects and updated path-planning strategies to account for the added scenario complexity.

References

[1]Pachter, M., Chandler, P., Purvis, K., Waun, S., Larson, R. "Multiple Radar Phantom Tracks From Cooperating Vehicles Using Range-Delay Deception." World Scientific, 4th Annual International Conf. on Cooperative Control and Optimization, Destin, FL, Nov. 2003 (to appear).

[2]Stimson, George W. "Introduction to Airborne Radar" 2d Ed. SciTech Publishing, Raleigh, NC, 1998.

[3]Vakin, S.A., Shustov, L.N., Dunwell, R.H. "Fundamentals of Electronic Warfare." Artech, Norwood, MA, 2001.

[4]Dubins, L. E. "On Curves of Minimal Length with a Constraint on Average Curvature, and with Prescribed Initial and Terminal Positions and Tangents." American Journal of Mathematics, Vol. 79, 1957. pp. 497-516.

[5]Bryson, A.E. Jr. "Dynamic Optimization." Addison-Wesley; Menlo Park, CA; 1999.

[6]Bryson, A.E. Jr. and Ho, Y. "Applied Optimal Control (Revised Printing)." Hemisphere Publishing Co.; 1975.

[7]Dennis, J.E. Jr., Schnabel, R.B. "Numerical Methods for Unconstrained Optimization and Nonlinear Equations." Society for Industrial and Applied Mathematics; Philadelphia, PA; 1996.